

Open Research Online

The Open University's repository of research publications and other research outputs

AM-OER: An Agile Method for the Development of Open Educational Resources

Journal Item

How to cite:

Arimoto, Mauricio M.; Barroca, Leonor and Barbosa, Ellen F. (2016). AM-OER: An Agile Method for the Development of Open Educational Resources. *Informatics in Education*, 15(2) pp. 205–233.

For guidance on citations see [FAQs](#).

© 2016 Vilnius University

Version: Version of Record

Link(s) to article on publisher's website:

<http://dx.doi.org/doi:10.15388/infedu.2016.11>

Copyright and Moral Rights for the articles on this site are retained by the individual authors and/or other copyright owners. For more information on Open Research Online's data [policy](#) on reuse of materials please consult the policies page.

oro.open.ac.uk

AM-OER: An Agile Method for the Development of Open Educational Resources

Maurício M. ARIMOTO^{1,2}, Leonor BARROCA³, Ellen F. BARBOSA¹

¹*Institute of Mathematics and Computer Sciences, University of São Paulo – ICMC/USP, Brazil*

²*Centre of Technological Sciences, State University of Northern Paraná – UENP, Brazil*

³*Faculty of Mathematics, Computing and Technology, The Open University, United Kingdom*
e-mail: mauricioarimoto@gmail.com, l.barroca@open.ac.uk, francine@icmc.usp.br

Received: March 2016

Abstract. Open Educational Resources have emerged as important elements of education in the contemporary society, promoting life-long and personalized learning that transcends social, economic and geographical barriers. To achieve the potential of OERs and bring impact on education, it is necessary to increase their development and supply. However, one of the current challenges is how to produce quality and relevant OERs to be reused and adapted to different contexts and learning situations. In this paper we proposed an agile method for the development of OERs – AM-OER, grounded on agile practices from Software Engineering. Learning Design practices from the OULDI project (UK Open University) are also embedded into the AM-OER aiming at improving quality and facilitating reuse and adaptation of OERs. In order to validate AM-OER, an experiment was conducted by applying it in the development of an OER on software testing. The results showed preliminary evidences on the applicability, effectiveness and efficiency of the method in the development of OERs.

Keywords: open educational resources, learning design, agile methods.

1. Introduction

Open Educational Resources (OERs) have been gaining importance for promoting life-long and personalized learning, helping to break demographic, economic, and geographic educational boundaries (Yuan *et al.*, 2008). OERs refer to digital materials provided freely and openly for educators, learners/self-learners to be (re)used with the purpose of teaching, learning and research (OECD, 2007). They range from lecture notes, images or audio and video files up to course materials, full courses and textbooks, as well as any software or other tools, materials or techniques used to support the construction and access to knowledge.

OERs open new possibilities for the production and dissemination of knowledge, while promoting an adaptive learning environment, suitable for each individual needs. Open educational practices, especially those based on the construction and adoption

of OERs, offer opportunities for innovation on different levels and learning modalities (face-to-face, blended or distance learning) with significant impact on education.

Despite the transformative potential of OERs to facilitate the access to knowledge and improve the education at a global level, many issues are still challenging. One of the challenges faced by developers and practitioners of OERs has been how to produce quality and relevant materials that can be reused and adapted in different contexts and learning situations. In addition, educators have difficulty in understanding the implicit design behind OERs and how to reuse and adapt them to their own teaching context (Dimitriadis *et al.*, 2009). The design and quality of available OERs have been hindering their potential for (re)use, also constituting issues that need more attention (OPAL, 2011; Conole, 2013).

In a different but related perspective, Learning Design (LD) has developed as a means of helping teachers make informed choices in terms of creating pedagogically effective learning interventions that make effective use of technologies. Research on LD has increased in the last few years primarily due to the gap between the potential and current use of technology to support teaching and learning (Conole, 2013).

In the context of OERs, studies have suggested the need for systematic and, at the same time, flexible approaches to support their design, creation and sharing (Sclater, 2009; Conole *et al.*, 2009; Arimoto *et al.*, 2014). In most initiatives related to the delivery of OERs, the production activities are the most costly, highlighting the need for an effective process for a sustainable delivery of OERs (Schuwer *et al.*, 2010). However, initiatives to foster the design and creation of quality OERs with reduced time and costs are still incipient. There is no evidence in the literature of approaches to effectively support the development of OERs (Arimoto and Barbosa, 2012).

This paper offers a contribution to address this need, by establishing an agile method for the development of OERs named AM-OER. The method learns from Software Engineering practices, particularly agile methods Scrum (Schwaber, 1995) and eXtreme Programming (XP) (Beck and Andres, 2004). Scrum and XP have different but complementary approaches. While the focus of the Scrum is on the planning and project management, XP focuses on the software development practices. Both focus on collaborative and flexible aspects of development and improvement of quality and productivity, which are concerns regarding the development of OERs.

AM-OER also embeds practices of LD that originate in the Open Learning Design Initiative (OULDI) project (Brasher *et al.*, 2012), proposed by the UK Open University. We incorporate practices of LD into the AM-OER that considers not only LD as dynamic process but also allows for the design to evolve incrementally, and be modified and enhanced as needed. The idea is to facilitate the reuse and adaptation of OERs and to contribute to their quality by embedding pedagogical design practices. An empirical study was performed in order to validate AM-OER by applying it in the design and creation of an OER on software testing.

This paper is organized as follows. In Section 2 we summarize some proposals related to this work; In Section 3 we describe the main characteristics and related practices, roles, phases and activities of AM-OER; In Section 4 we report an empirical assessment study to validate AM-OER by designing and creating an OER on software testing; In Section 5 we summarize the analysis and results of the experiment conducted; concluding remarks and future work are presented in Section 6.

2. Related Work

2.1. Development of OERs

There are several approaches for the development of learning materials in general. Most of them are based on traditional approaches derived from software development. Examples of such approaches include the methods for the development of Learning Objects (LOs) (Kemczinski *et al.*, 2012; Braga *et al.*, 2013). These approaches are plan-driven and have characteristics in common, with little flexibility and lack of involvement of users (especially educators and learners) throughout the development. Due to their strictly sequential characteristics, the development process becomes little flexible and adaptable to changes. In general, the team can only advance to the next stage after finishing the previous one.

There are initiatives that intend to make the development of LOs more collaborative and flexible. Boyle *et al.* (2006) proposed an agile method for LOs, covering the main phases of the development process. Similarly, Lapolli *et al.* (2010) proposed a model based on agile practices focusing on the content and instructional design.

The approaches aforementioned are focused only on the development of LOs, and may not be suitable for the context of OERs. Little attention is given specifically to the development of OERs. Patricia *et al.* (2010) proposed a life cycle for the development of OERs with a set of well-defined phases based on ADDIE model (Molenda, 2003) that follows the traditional approach for software development, therefore, having the same shortcomings of the approaches previously mentioned.

Finally, particularly regarding to the use of agile methods in the context of OERs, Masson and Udas (2009) discuss the use of principles and practices of agile methods for enabling adoption and management of OERs. However, the authors have proposed an agile approach only focused on management of OERs.

2.2. Learning Design

The Open University Learning Design Initiative (OULDI) was initiated in 2007 to derive a more “practice-focused approach” for LD. It includes a set of practices and artifacts to represent design:

- Macro-level (the course map view): provides an overview of main components for the course, enabling educators to think about the design of a course based on four dimensions: contents and experience; (2) guide and support; (3) communication and collaboration; and (4) reflection and demonstration.
- Meso-level (the learning outcomes view): shows how learning activities and assessment tasks are linked with the intended learning outcomes of the course.
- Pedagogy profile: articulates type of activities in which learners undertaken during the course, categorized as assimilative, information handling, communication, productive, experimental, adaptive and evaluation.

- Course dimensions: provide more information on the nature of the course and how it is supported. It can be seen as a refinement of the course map view.
- Micro-level (the task swimlane view): maps tasks that the learners undertake to the contents and tools they use during the activities in the course.

Although the OULDI approach intends to make the design more explicit, it does not specify the steps and guidelines for a LD process for the design of OERs. Besides that, a LD approach cannot be seen as a set of separate levels, often requiring refinement and improvement. This implies that the design should allow moving back and forwards to the process according to the needs (Avraamidou and Economou, 2012).

Our proposal focuses on the whole development process of OERs. It combines agile practices and LD practices to promote flexibility and productivity during the development, and to allow designing and creating pedagogically effective OERs.

3. AM-OER Overview

In Fig. 1 we illustrate the AM-OER development flow and its general characteristics and practices (discussed in Section 3.1). The OER development project starts from the learning needs and/or problems identified by the educators. Then, a high-level initial architecture with the main components needed to the OER is sketched. Team gets together to plan the OER releases (small deliveries of OER) and the sprints (predefined period of time) needed to develop them. They define which parts (modules) of OER will be designed and created firstly, in the current sprint. Modules are developed iteratively and incrementally by short sprints, with a sprint lasting from days to few weeks. During the

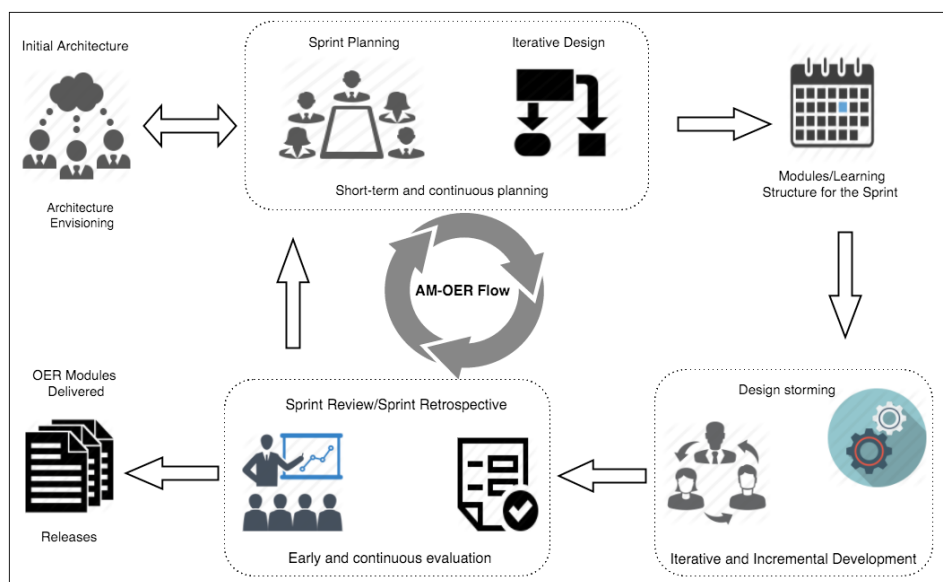


Fig. 1. AM-OER overview.

sprint, team members communicate and constantly interact with each other to discuss the development activities, monitor their progress, and to identify the main obstacles that hinder the progress of work.

At the end of the sprint, the team gets together to evaluate the produced release and the progress of the sprint as a whole, identifying changes, improvements and needs for new components. Small releases of the OER approved by the educators are delivered in a short period of time to be used by learners, obtaining important feedback about their learning experiences on the OER. The process is repeated until all planned releases are developed and a final version of the OER is delivered (final and updated release).

Next, we describe the AM-OER method, focusing on its main characteristics and related practices, roles and phases.

3.1. *Characteristics and Practices*

AM-OER has a set of characteristics and practices derived from Software Engineering (Ambler, 2001; Schwaber, 1995) and LD (Conole, 2013). Collectively implemented, they effectively assist and guide the process of OERs development.

In Table 1 we summarize some agile practices we incorporated into the AM-OER. Following we describe how these practices were adapted and combined with LD practices to support the design and creation of OERs.

Table 1
Agile practices

Practices	Description
Small releases	Releases include small set of features prioritized by the customer based on their relevance.
Sprint planning	The team reviews and decides on what will be implemented within a sprint and how the will be performed
Architecture envisioning	Initial software architecture and requirements are designed at the beginning of a project to identify and think about critical issues.
Iterative modeling/design	Software functionalities are designed at the beginning of iteration to identify team's strategy for that iteration.
Model/design storming	Software functionalities are designed on a Just-inTime (JIT) basis to reflect on specific aspects of team's solution.
Simple design	A solution is designed as simple as possible to adapt to changes, without demanding much effort.
Refactoring	Small changes are performed to improve part of a solution without changing its semantic meaning.
Continuous integration	Releases of software are integrated and tested several times a day, whenever a task is completed.
Collaborative development	Team members constantly interact and communicate throughout the development process, promoting a collaborative and productive environment.
Sprint review	Team gets together at the end of sprints to demonstrate the work done within the sprints.
Sprint retrospective	Team gets together after the sprint review to discuss what working and what is not working well in the project in order to improve the next sprint.

3.1.1. *Short-term and Continuous Planning*

The OER project focuses on short-term and continuous planning in each development cycle. Firstly, the team gets together to plan and agree about the **small releases** to be delivered in a short period of time, ranging from a week or more to months. In the context of OERs, a release consists of a set of modules or small modules (parts of units of learning) composed by learning contents, learning activities, assessments, roles and tools needed to the intended learning.

The team plans the sprint(s) needed to develop the releases. Each release is divided into one or more sprints. Sprints are shorter, ranging from days to few weeks. The team conducts the **sprint planning** to decide on what will be delivered at the end of the current sprint, and on the way in which the work will be conducted. Then, the team members select the amount of work that each one can develop within the sprint. At the end, it is possible to obtain a detailed plan of what will be developed within the sprint.

3.1.2. *Dynamic and Incremental Design*

The design process of OERs is dynamic and involves the whole team, especially educators and learners. By using the term “educators” we also include practitioners, teachers, lectures and tutors. The design is performed incrementally, allowing modifications and improvements be carried out in any part of the process. The design involves design just enough for the current sprint, prioritizing the most relevant aspects to be addressed by the OERs under development (**simple design**) (Ambler, 2001). OERs are designed to facilitate the understanding about how contents, activities, roles and tools associated to the OERs should be connected and interact with each other to contribute to a more effective learning.

Architecture envisioning is adapted to the context of OERs at the beginning of the project (known as sprint #0) to obtain an overall structure of the intended OER. LD practices (Conole, 2013) are used together to help educators and potential learners to think about and define the key components of the intended OER, including: (1) main learning contents needed; (2) context and application domain; (3) learning assessment activities; and (4) learners/educators dialogical and collaboration issues.

Iterative design is adapted to the context of OERs at the beginning of a sprint (sprint #0) to sketch iteratively and incrementally the learning structure for the OER. LD practices are used together to describe and map the main components of the intended learning structure, including learning activities and their connections to the intended learning outcomes, the contents, the tools and assessments.

Design storming is adapted to the context of OERs before creating related contents to sketch specific aspects of the OER activities flow. LD practices are used together to describe and map how learners will interact with their individual activities and tasks, specific contents or resources/tools, and assessments. It triggers refining and decomposition of activities from learning structure into simpler individual activities and tasks, helping educators to reflect upon an aspect of design solutions.

3.1.3. *Iterative and Incremental Development*

The development of OERs occurs incrementally by short and repetitive sprints. The iterative and incremental characteristic provides a flexible approach, in which responses to changes can be made faster during the process to minimize their cost and impact.

Releases are delivered periodically throughout the process and to be used by a target audience. The aim is to ensure that learners have the opportunity to use and evaluate them, providing rapid feedback on the OER. New learning contents and enhancement requests are produced by the team as development progresses. By using **refactoring**, the team can review and improve the OER under development through small changes without changing the intended learning outcomes. This should be performed whenever an opportunity for change and/or improvement is identified during the development.

Modules of OER are integrated and validated throughout the process (**continuous integration**) by the team. At the end of all sprints planned for the development of OER, a stable and robust release is delivered.

3.1.4. *Collaborative Approach*

All those involved in the OERs development should interact and cooperate constantly throughout the development. This helps to solve problems related to the specification, design and implementation more quickly, reducing time and effort and contributing to a more effective development process of OERs.

The **active participation of users** (educators and learners) is encouraged throughout the process, either in person or via collaborative technologies (Web 2.0 tools such as wikis, microblogging, social networking and instant messaging systems). They assist in the establishment of the goals of OER, together with the learning objectives, learning activities and pathways, contents, tools and assessments.

During the OERs development, several activities are carried out in groups through brainstorming and workshop sessions (such as initial architecture envisioning, sprint planning, iterative design, design storming, sprint review, and sprint retrospective), either face-to-face or by synchronous communications (text mode or videoconference). **Collaborative development** is promoted, increasing the interaction and communication amongst all involved in the development.

3.1.5. *Early and Continuous Evaluation*

Evaluation is carried out early and continuously throughout the development process. During the **sprint review**, at the end of the sprint, educators (and learners) have the opportunity to check whether the OER being created is in agreement with those previously planned before the sprint starts. It is possible to identify new perspectives for the OER, possibility of modifications and/or inclusion of new learning contents and needs for improvements in a short period of time, minimizing the cost of change.

During the **sprint retrospective**, the team members have the opportunity to think about how the OER development is progressing. They can identify and analyze the problems and mistakes occurred during the process. Lessons learned help them to decide what can be done to improve the performance of next sprints.

3.2. Roles

The AM-OER team should have multidisciplinary characteristics and bring together the skills needed for agile development of OERs. AM-OER considers four roles:

- (1) User.
- (2) Intellectual Property (IP) Expert.
- (3) Coach.
- (4) Development Team.

The **User** collaborates with the development team, pointing out the problems, needs and key characteristics for the OER, having associated roles: (1) **Educator**: he/she provides support for determining relevant components of OER. Plays a fundamental role as a creator and consumer of OER, helping to develop and validate the contents and activities; and (2) **Learner**: uses the OER during the development process and evaluates whether the learning objectives are being met. He/she is encouraged to get involved throughout the OER project to obtain feedback early and guide the development.

The **IP Expert** is responsible for identifying and establishing which persons and institutions have intellectual property rights on the OER. He/she plays an essential role especially when the OER is derived from third-part materials, maintaining clearly defined the intellectual property rights.

The **Coach** detains global knowledge on the whole development process. He/she works closely with users and development team, serving as a communication channel between them. He/she is responsible for managing the progress of the project, solving/minimizing the conflicts, problems and obstacles faced by team.

The **Development Team** designs and creates the contents and activities associated to the OER such as content editing, sequence of learning activities and integration of multimedia components. It can have associated roles: (1) **Designer**: he/she models the contents and their relationship with learning activities, tools, roles involved in the learning intervention, and the learning outcomes; (2) **Media creator**: he/she selects and creates the appropriate media to the context of OER, ensuring their accuracy, consistence and organization; and (3) **Reviewer**: he/she implements and executes the validation and verification activities to ensure the quality of OER.

3.3. Phases

The life cycle proposed to support the agile development of OERs follows the iterative and incremental characteristic of agile methods. Sprints are shorter to promote visibility for the OER under development, i.e., an opportunity for educators perceive how the development of OERs is progressing during the cycle. The aim is to provide the educators with mechanisms for monitoring the OERs development. Besides that, a small release of OER is incrementally delivered in relatively short period of time to be used by learners, obtaining important feedback about their learning experience, which can be used to improve and incorporate new components to the OER.

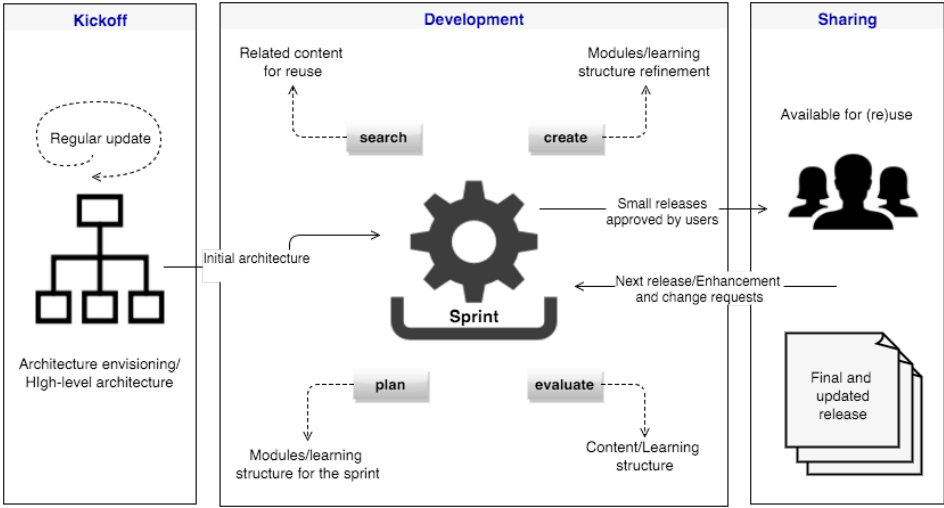


Fig. 2. AM-OER lifecycle for OERs.

The AM-OER life cycle for OERs is flexible, allowing the team to make changes, if necessary, during the development of OERs. The life cycle is divided into three phases (Fig. 2):

- (1) Kickoff.
- (2) Development.
- (3) Sharing.

3.3.1. *Kickoff*

This phase comprises the beginning of the project (sprint #0), wherein users (educators and learners) come together with the development team to identify and discuss on the learning needs and/or problems to be addressed by the OER and its overall structure. In Table 2 we summarize the steps of this phase, including the main participants,

Table 2
Kickoff phase: summary of the steps

Steps	Participants	Input/output Documents	Outcomes
Identify learning needs	Whole team		Objectives
Establish learning contents	Educators/ Development team	Objectives	Main contents established (Initial architecture view)
Establish learning assesments	Educators/ Development team	Objectives	Assessment activities (Initial architecture view)
Establish means of communi- cation and collaboration	Educators/ Development team		Means of communication and collaboration established (Initial architecture view)

the input and/or support documents, and the outcomes and/or artifacts. The steps are described next.

Boyle *et al.* (2006) argue that a common deficiency of many learning materials is the inadequate analysis of the learner's needs. Then, we must focus on the needs and problems that learners face. In this case, the educator plays a crucial role as a source of information, since he/she has expertise on the student's background and on the knowledge domain.

To help in identifying the learning needs/problems, some issues should be considered:

- What are the main needs and/or motivations for developing an OER?
- What are the problems to be addressed by the OER to meet the learning needs?
- Who is the target audience (for instance, informal learning, primary and secondary school, undergraduate and graduate, and lifelong learning)?
- What is the learning context and application domain of the intended OER?

Educators start by establishing the general goals of the OER, i.e., what they intend to cover/present in the OER to be developed. For instance, the general goal of an OER could be: (1) to explain the fundamentals of functional (black-box) testing technique; or (2) to provide students a general introduction to testing in concurrent programs.

Educators sketch the learning objectives of the OER, consisting of specific statement of teaching/learning intention. DeVries (2013) highlights that many OERs do not have basic elements of LD such as learning objectives. This makes it hard to assess the OER in terms of its overall purpose and learning outcomes, and of the pedagogical alignment of learning materials, activities and assessments. He also argues that learning objectives are essential elements for reuse, help identifying if an OER has the level of coverage and depth appropriate to be used in a different context or learning situation.

Bloom's taxonomy (Bloom *et al.*, 1956) is useful for defining learning objectives. Such taxonomy intends to organize the acquisition of cognitive skills/learning into levels (i.e., Knowledge, Comprehension, Application, Analysis, Synthesis, and Evaluation), facilitating the identification and measurement of the learning objectives.

Examples of learning objectives statements may be: (1) students would understand the main ideas behind testing on concurrent programs; or (2) students will be able to plan and implement test cases in concurrent programs using functional testing.

Educators also need to specify the context or domain in which the OER will initially intended to be applied. The context should also address cultural and languages issues. For instance, learning materials in English are very common worldwide, but cultural context of learning usually differs from one country to another. Then, these issues should be considered during the design and creation of OERs.

After established the learning needs, the team discusses about its overall structure. The main components of the OER are sketched, but without too much detail, following the concept of **architecture envisioning** (Ambler, 2001), as the design should be constantly evolving throughout the sprints. Practices of LD (Conole, 2013) are used together to identify and map the main components of the OER including the contents

needed, the ways of learning assessment and how learners will interact with educators and colleagues. An Initial Architecture View is created based on the Course Map View of OULDI (Brasher *et al.*, 2012). The Initial Architecture View is used as input for the next phase.

The contents to be developed and integrated in the OER should be established in a high level, without going into details on the type of content or media (for instance, whether it will be an open textbook, a research paper, a podcast or a video) to be used. There is no need to identify all contents a priori; further contents can be added or changed later. Pre-requirements or expertise needed to use the OER should be also specified, including a calendar with information about the duration related to its usage in a course or training.

To help in establishing the contents of the OER, some issues should be considered:

- How will OER modules be delivered to learners (face-to-face, online or hybrid)?
- How will learners be supported (face-to-face, online or hybrid)?
- How will subjects, topics or concepts be introduced to learners?
- What kind of activities learners will need to perform?

Examples of learning contents may include: lessons, lab activities, guidelines, readings, examples, support materials, case studies, pilot projects, among others.

Educators need to think about and establish how learners will be evaluated within a learning intervention. Learning assessment activities help educators gather evidence from learners during a class/course to adjust and improve their teaching strategies. In the same way, the learners can adjust and improve their learning strategies according to the learning assessment activities. To help in the definition of the learning assessment activities, educators should consider the following issues:

- How will assessment activities be (online, paper based or both)?
- Which assessment strategy will be used (diagnostic, formative, summative or all of them)?

Examples of learning assessment activities may be: in-text questions, self-assessment questions, oral or written presentations, reports, essays, among others.

Finally, educators need to think about and define the dialogical and collaboration aspects related to the usage of OER. For this, some issues should be considered:

- How will learners communicate and collaborate with their colleagues (online, face-to-face, both)?
- How learners will communicate and collaborate with educators (online, face-to-face, both)?
- How learners will perform their activities (individually, peer-to-peer-work, work in a group)?

Examples of means of communication and collaboration may include: instant messaging system, chat, social networking, mailing list, brainstorming sessions, peer-to-peer works, work in groups, among others.

At the end of this phase, the whole team needs to agree on the general goals, the learning objectives and the Initial Architecture View of the desired OER. Then, the new OER is approved and can be developed in the next phase.

3.3.2. Development

The development phase comprises the whole effort to plan, search, create and evaluate the OER within a sprint. In Table 3 we summarize the steps for planning, designing, creating and evaluating an OER, including the main participants, the input and/or support documents, and/or the outcomes or artifacts. The steps are described next.

Before starting the development cycle (sprint #1), the whole team comes together through a workshop session to conduct the **sprint planning** under the supervision and mediation of Coach. Small modules of the OER should be prioritised and selected to be developed during the current sprint based on the Initial Architecture View.

Firstly, the team should establish a plan for a release to be delivered to a target audience. Depending on the size of project, an OER can have one or more releases and each release can need one or more short sprints.

Educators should prioritize the small modules of the OER and introduce to the designers and media creators, indicating what should be done with each one. They should focus on the most relevant aspects to be addressed by the OER. Other secondary aspects or less important should be discussed later during the development. Then, the development team should select the modules to be created in the current sprint taking into account the priority of the OER modules previously established.

The development team can make estimatives to predict the effort to create or repurpose an OER, or to predict the duration of the OER project as a whole.

The OER is designed iteratively in small increments (**iterative design**). By incorporating practices of LD, a lightweight design for learning activities, content/tools, assessment/learning outcomes and their relationship are sketched out, enabling team to think about and define strategies for each sprint. The result produces the Learning Structure View for the current sprint, based on the Learning Outcomes View of OULDI (Brasher *et al.*, 2012). The Learning Structure View is a mandatory artifact used, updated and integrated throughout the development.

Table 3
Development phase: summary of the steps

Steps	Participants	Input/output Documents	Outcomes
Prioritize and select OER modules	Whole team	Initial architecture view	OER modules prioritized and selected
Design the learning structure	Whole team	Objectives/ Initial architecture view	Learning structure view
Define metadata	Educators/ Development team	Initial architecture view	Primary metadata
Search learning contents	Educators/ Development team	Objectives/ Initial architecture view	Related contents
Evaluate and adapt learning contents	Whole team	Related contents	Related contents approved
Create OER modules (from scratch)	Development team/ Educators	Objectives/ Initial architecture view	OER modules created
Establish licensing policies	Educators/IP expert	OER modules created	Licensing chosen
Evaluate OER modules/sprint execution	Whole team	OER modules created	OER modules approved/ Lessons learned

Educators and the development team should start identifying and gathering the primary metadata for the OER module. Metadata describe relevant characteristic of the OER, facilitating their reuse and recovery by search engines. When an OER has integrated metadata, any user can easily find it (Madden, 2010). The metadata specification must be standardized (such as Learning Resource Metadata Initiative: lrmi.net/about/) to facilitate the reuse and discoverability of OER.

The team members should check for related contents to be reused to compose the OER. Levery (2012) argues it may be easier or more efficient to reuse and adapt existing contents than creating new ones. There are many places where contents can be searched. Materials from educators or colleagues can be used as starting point. On the other hand, the Internet provides facilities for seeking online contents through different mechanisms including specialized search engine (e.g., OpenCourseware Consortium: ocwconsortium.org), institutional repositories (e.g., ConneXions: cnx.org), digital libraries (e.g., OER Commons: oercommons.org), and many others.

While searching for related contents, specific issues need to be taking into consideration. An essential issue refers to the contents accessibility. Materials should be accessible and available in file formats that do not difficult their modification and conversion to other formats. Therefore, when searching for and selecting related contents some issues should be taken into consideration:

- Consider different search engines, databases and repositories. Some search engines incorporate a large number of databases; while others are dedicated to seeking specific media (open textbooks, presentations, images, audio/video). Be aware on the use of relevant keywords related to the subject. This should avoid that potentially relevant contents be not located by the search engines.
- Prioritize contents provided with little or no restriction regarding their (re)use and adaptation. Verify if the file formats used can be modified and tailored to the desired needs.
- Be aware on the licensing types used on the available contents. Some licenses do not allow modification or remix on the contents; while others require that reused materials be shared under the same license used by their owners.
- Make sure that the contents are from reliable sources, including institutions/organizations engaged with education, and renowned authors. Verify if the contents are associated with ranking system, and if they are recommended by users. This is a means of ensuring the quality of the contents.

After obtaining the contents, educators and reviewers need to evaluate whether they are appropriate for the OER purpose, before the development team modify, re-contextualize or adapt them. Some issues should be considered:

- (1) Analyze whether the contents found fit the didactic and pedagogical objectives of the OER.
- (2) Identify what are the weakness and shortcomings of the contents or whatever needs to be revised, changed or improved.
- (3) Define what has to be added or mixed with the contents, and what has to be created from “scratch” to compose the intended OER.

In many cases, it is necessary to develop an OER from “scratch”. In this context, the major difficulty is related to the reuse of third-part learning materials to produce something new. Regardless of reuse third-party material or not, an OER is designed and created by the development team through several and repetitive sprints, culminating with the delivery of **small releases** at the end of a sprint or small set of sprints.

Before starting to create the related contents, the team comes together with the **active participation of users** to create a JIT design (**design storming**). By incorporating practices of LD, the development team members refine and decompose the OER activities in simpler activities and atomic tasks, helping them to reflect upon one aspect of the design solution and how they can transform it for a more effective OER with embedded pedagogical design practices. The result culminates with the Learning Task View based on the Task Swimlane View of OULDI. This artifact maps individual learning activities to the components that need to be addressed by such activities:

- (1) Contents or tools involved in the activities.
- (2) Types of tasks to be performed by learners.
- (3) Roles of everyone involved in the learning intervention.

The development team creates the OER activities and contents which will compose the OER module, including any kind of content and activities such as open textbooks, html pages, lecture notes, audios/videos, images, animations, tests, among others.

The team members work in constant collaboration throughout development. New solutions and improvements could be highlighted through feedback provided by interactions and cooperations with educators and learners. The OER is reviewed and updated throughout the development. The team members constantly refactor (**refactoring**) their solution aiming to simplify and enhance it. They often integrate and link (**continuous integration**) all components that compose the OER produced.

The IP Expert is responsible for establishing the licensing policies to share the OER with others. OER implies the use of open licenses with little or no restriction. Open license enables sharing an OER without payment of a royalty or license fees.

When establishing the licensing policies of the OER the IP Expert should: (1) verify the authorship and intellectual property rights of third-part materials (when used); (2) decide how the OER will be available (for instance, if non-commercial use of work is allowed or not and if adaptations of work is allowed to be shared or not); and (3) choose the appropriate license taking into account items 1 and 2.

At the end of the sprint, the whole team comes together during the **sprint review** to discuss, evaluate and approve/disapprove the OER module produced within the sprint.

There are different characteristics which can be considered during an assessment program. In order to assist in the OERs evaluation we proposed a set of assessment criteria/issues based on previous work (International Organization for Standardization, 2001; Leacock and Nesbit, 2007; JISC, 2010), classified into three categories:

1. **Didactic-pedagogical**, such as: (1) accuracy of the contents for the learning objectives; (2) relevance of the contents/fits for purpose; and (3) pedagogical design of materials.
2. **Technical**, such as: (1) interoperability; (2) accessibility; (3) usability; (4) discoverability; and (5) localisation/globalisation capabilities.

3. **Legal**, such as: (1) intellectual property and rights; and (2) open licenses policies for the contents.

We highlight that the criteria aforementioned can be used together or separately, depending on the purpose of the evaluation. Other criteria should be also defined and adopted to support the evaluation.

A set of activities should be conducted to evaluate and approve the work and artifacts produced in the current sprint. Different approaches should be considered such as: (1) users assessment, conducted by educators/specialists, learners, and others users; (2) peers review, conducted by academic staff, educators/specialists and others engaged with education; (3) individual assessment, conducted by educators/specialists within the team or institution/organization; and (4) external assessment, conducted by educators/specialists outside the team or external institution/organization.

Educators and learners (external reviewers can participate) are involved in verifying whether the learning pathways associated with the content contribute to the learning. Educators also analyze whether the type of content and activities, assessments, and tools are appropriate to the purpose of the OER, e.g., if they are aligned with the intended learning outcomes. The results gathered from the evaluations should be used as the basis for proposing changes and adding improvements for the next release of OER.

After the sprint review, the team come together to conduct the **sprint retrospective** to evaluate the overall sprint execution. The team should identify “what worked well” and “what did not work well” within the sprint. Then, the team should discuss about “what needs to change and improve” in the next sprints. Lessons learned and feedback from the evaluation should be analyzed, gathered, and used for changing and improving the following sprints, contributing to the continuous improvement process.

3.3.3. *Sharing*

The OER release should be provided to be used in a learning environment, and effective access to the release should be allowed according to its context. In Table 4 we summarize the steps of this phase, including the main participants, the input and/or support documents, and the outcomes and/or artifacts. Such steps are described next.

The OER release is delivered for use in a learning environment with a certain group of learners. This is critical to identify weaknesses and propose improvements. Educa-

Table 4
Sharing phase: summary of the steps

Steps	Participants	Input/output Documents	Outcomes
Deliver to a target audience	Educators	OER release	Learner's experience
Make available in easily accessible locations	Development team	OER release	Available on platforms/repositories or stand-alone wikis/websites
Gather feedback and review	Educators/ Development team	Released OER	Learner's experience/User's feedback/Weaknesses and shortcomings

tors need to provide support to learners in their activities and monitor their progress and achievements. Data about the learner's experience should be collected and analyzed to improve the quality of the OER release.

Effective access to the OER should be through platforms, repositories, and institutional or stand-alone websites, including associated metadata and license with few or no restriction on its (re)use. Web 2.0 technologies and social software can be used as support such as, microblogging and social networking. Tools such as, Slideshare and Flickr, are accessible and easy mechanisms for sharing OERs. Other tools such as RSS Feeds are also useful, particularly to link related contents, as well to disseminate and facilitate access to the OER. The OER should also be available in CD/DVD formats, especially for users with Internet access limitations. The widely availability of OER allows other users (such as external educators and researchers) to reuse and adapt it to their own context and needs.

Educators should gather feedback from learner's experience on the use of OER in a learning situation, as well as the feedback provided by others users. Possible changes and improvements should be analyzed and reviewed by the team and may be implemented in the further release.

Additionally, the team should review the OER release in order to identify mistakes, weaknesses and shortcomings. This includes the need to fix reported problems, perform small changes and adaptations, promote improvements or add new contents, activities, supporting materials or tools.

The AM-OER development cycle should be restarted whenever changes and adjustments are required to keep the OER up-to-date and improve its quality.

4. Applying AM-OER in the Software Testing Domain: An Experiment

In this study we describe an explanatory study based on experimentation. We conducted an experiment involving the development of a testing course to assess the applicability, effectiveness, efficiency and quality of the results produced by the usage of AM-OER.

Experiment represents an action to discover something unknown or testing a hypothesis involving the collection and analysis of data to provide "meaning" to the obtained data (Basili and Forest, 1999). We adopted an experiment to allow a more rigid control on the environment, and a more rigorous manipulation of the phenomenon we study. It can generate more concise results based on quantitative analysis, providing evidence of the validity of AM-OER to develop OERs. It can also allow the generalization of the results within a population, and the replication of the experiment.

4.1. Objectives

The overall objective of this experimental study was to assess the applicability, effectiveness and efficiency of AM-OER by comparing it with the AD-HOC method. In the

AD-HOC method, the development is informal, with no defined process and with no guidelines or instructions to guide the development of OERs.

To achieve the objective above we propose the following Research Questions (RQ):

- RQ1. How effective is AM-OER in the development of OERs compared to the AD-HOC method?
- RQ2. How efficient is AM-OER in the development of OERs compared to the AD-HOC method?
- RQ3. How much better are the results obtained by AM-OER compared to the AD-HOC method?

4.2. Hypotheses and Variables

Hypotheses are testable propositions made about a phenomenon we intend to test. They are formulated in null and alternative hypotheses. Null hypotheses (H_0) are the hypotheses we intend to reject:

1. The effectiveness of AM-OER is similar to the effectiveness of AD-HOC method $\rightarrow H_0: \pi_{AM-OER_{Effe}} = \pi_{AD-HOC_{Effe}}$.
2. The efficiency of AM-OER is similar to the efficiency of AD-HOC method $\rightarrow H_0: \pi_{AM-OER_{Effi}} = \pi_{AD-HOC_{Effi}}$.
3. The quality of the results of AM-OER is similar to the quality of the results of AD-HOC method $\rightarrow H_0: \pi_{AM-OER_{Qual}} = \pi_{AD-HOC_{Qual}}$.

Alternative hypotheses (H_1) are hypotheses we expect to assume as true:

1. The effectiveness of AM-OER is higher than the effectiveness of AD-HOC method $\rightarrow H_1: \pi_{AM-OER_{Effe}} \neq \pi_{AD-HOC_{Effe}}$ or $H_1: \pi_{AM-OER_{Effe}} > \pi_{AD-HOC_{Effe}}$.
2. The efficiency of AM-OER is higher than the efficiency of AD-HOC method $\rightarrow H_1: \pi_{AM-OER_{Effi}} \neq \pi_{AD-HOC_{Effi}}$ or $H_1: \pi_{AM-OER_{Effi}} > \pi_{AD-HOC_{Effi}}$.
3. The quality of the results of AM-OER is higher than the quality of the results of AD-HOC method $\rightarrow H_1: \pi_{AM-OER_{Qual}} \neq \pi_{AD-HOC_{Qual}}$ or $H_1: \pi_{AM-OER_{Qual}} > \pi_{AD-HOC_{Qual}}$.

The experiment configuration is shown in Fig. 3. The hypotheses are tested through independent and dependent variables. The independent variables or factors are the input data under control, and can be modified within the experiment. They include the methods used by both groups of subjects – the AM-OER and the AD-HOC methods.

Dependent variables are related to the results to be analyzed and interpreted after the execution of the experiment, being directly influenced by the independent variables. They include:

- (1) **Effectiveness:** measures the level in which a method can develop the planned OER.
- (2) **Efficiency:** measures the effort needed by each group of participants to develop and delivery the intended OER.
- (3) **Quality results:** measures the level in which a method achieves the percentage of compliance to quality attributes and desirable characteristics of the OER.

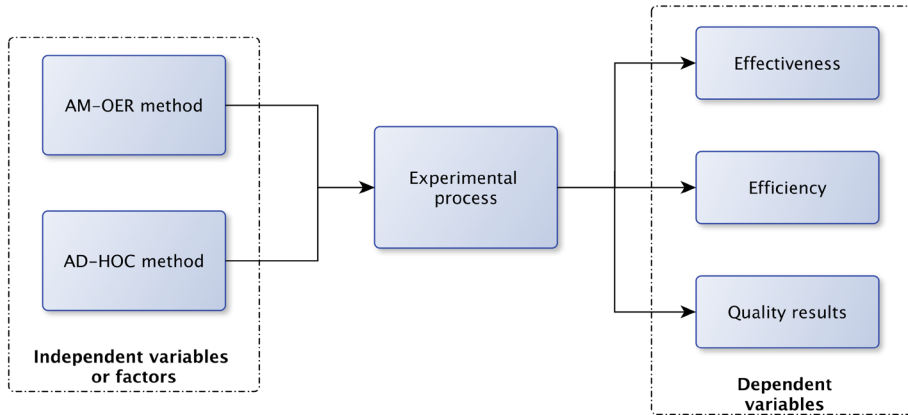


Fig. 3. Experiment configuration.

4.3. Subjects

We used a non-probability sampling by convenience to select the ten participants of the study. They include P.h.D students, educators and researchers from the Software Engineering and Computer applied to Education Groups of the Institute of Mathematics and Computer Sciences (ICMC) at University of São Paulo (USP).

The participants filled out a short questionnaire identifying their level of knowledge/experience in the topics related to the experiment. The results from experience were used to divide and compose the groups.

4.4. Experiment Design

We adopted a combination of “blocking” and “balancing” principles of design (Wohlin *et al.*, 2000). Blocking is used to minimise undesired effects of the experiment whilst balancing is used to facilitate and strengthen the data analysis.

To minimise the effect of “experience” of the participants, we grounded them into two blocks, trying to create homogeneous groups according to the experience of each participant. The two groups are balanced in terms of the number of participants as well.

The experiment was characterised as one factor and two treatments. It compares two treatments (AM-OER and AD-HOC methods) against each other in the development of an OER. Both groups develop the same OER related to the software testing domain, but each one uses a different treatment.

4.5. Statistical Methods

The sampled data was tabulated and graphically displayed, facilitating their visualisation and understanding. Descriptive statistics, including quartiles measures, were obtained to

give an overview of how the set of data was distributed. The sampled data was displayed graphically by box-plot to show the discrepancy between the data.

The next step was to test the set of hypotheses established for the experiment. By testing the hypotheses two types of errors can occur:

1. First, when the alternative hypothesis H_0 is rejected, but it is true. This type of error is characterised as Error I. The probability of this type of error can be evaluated as: $\alpha = P(\text{Error I}) = P(H_0 \text{ is rejected} \mid H_0 \text{ is true})$.
2. Second, when H_0 is accepted, but it is false. This type of error is characterised as Error II. The probability of this type error can be evaluated as: $\beta = P(\text{Error II}) = P(H_0 \text{ is not rejected} \mid H_0 \text{ is false})$.

From the aforementioned, we need to establish the levels of significance of the errors. In this experiment we adopted the usual practice of admitting a low value or level of significance for both errors: $\alpha = 0.5$, and $\beta = 0.5$.

To verify if the sampled data followed a normal distribution we used the Shapiro-Wilk Normality Test Shapiro and Wilk (1965), which is a method for normality test suitable for small set of data.

To test the hypotheses established for the experiment, we adopted the Chi-squared test Plackett (1983), a non-parametric method useful to compare proportions, i.e., the possible differences between the observed and expected frequencies for a certain event. In the R software (R Project, 1993) there is a function `prop.test()` for this purpose.

4.6. OER on Software Testing

The OER developed in the experiment was a software testing course, covering a brief overview of functional testing (black-box), and equivalence partitioning testing and boundary values analysis criteria (Myers, 2004). Each group had five hours to produce such OER. Following we summarize the main phases and activities for the development of such course by the group using the AM-OER.

4.6.1. Kickoff: Defining Objectives and Overall Structure for the Course

At the beginning of the project, the educator within the group specified the general goals and learning objectives of the course. The general goals were to provide students an overview of the functional testing, and show how such technique can be applied in the software construction. The learning objectives to be achieved at the end of the course were:

- Students will discuss and describe the fundamentals of functional testing.
- Students will be able to explain and distinguish the two major functional testing criteria: partitioning functional testing and boundary values analysis.
- Students will be able to apply the partitioning functional testing and boundary values analysis, and to argue and defend both criteria.

In Fig. 4 we show the Initial Architecture View, summarising the overall structure including the main components needed to the course. It includes the contents covered by

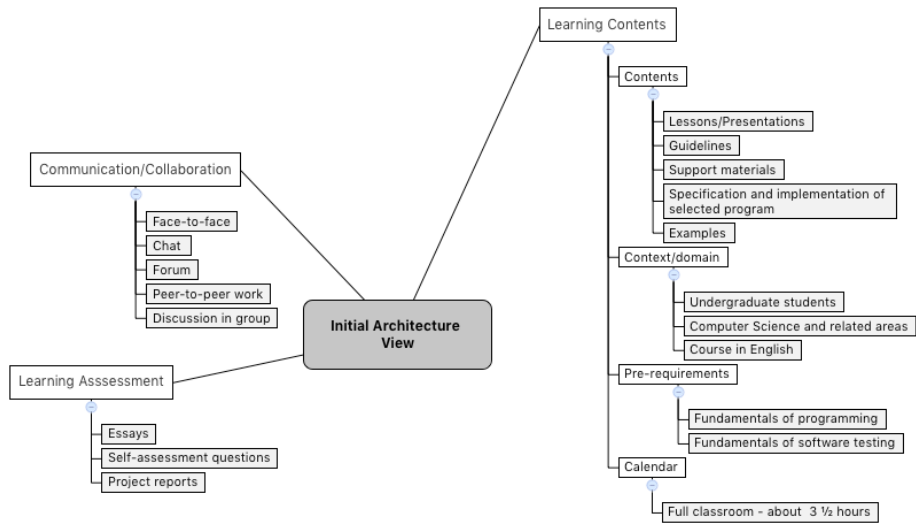


Fig. 4. OER on software testing: overall structure.

the course, the assessments criteria to evaluate learning, and the mechanisms supporting communication and collaboration during the learning intervention. This structure is specific to the AM-OER method, not being produced by using an AD-HOC method.

The target audience of the course includes undergraduate students in Computer Science and related areas such as Computer Engineering. The course is a full class within a software testing course with an expected duration of three hours and a half. Students are supported both physically and online. As prerequisites and experience, the students must have basic skills on fundamentals of programming and software testing.

Contents covered by the course include lessons, guidelines, examples, supporting materials and specification and implementation of a program named Cal¹. Students are evaluated through essays, self-assessment questions and project reports. They communicate and collaborate with colleagues and educator in person (face-to-face), or through chat, forum, peer-to-peer work and discussion in group.

4.6.2. Development: Designing, Creating and Evaluating the Learning Structure and Related Contents and Activities

The participants get together to discuss and think about the overall learning structure for the course. A Learning Structure View was created (Fig. 5), mapping the learning activities that students take part (such as *brainstorm the functional testing criteria*) with intended learning outcomes (such as *demonstrate knowledge and critical understanding on fundamentals of functional testing*), content (such as *Cal program specification/Cal program implementation*), tools (such as *framework xUnit*), and assessment activities (such as *essay summarising discussion*).

¹ This program is about a Calendar, wherein the user can obtain a calendar for a particular month and/or year.

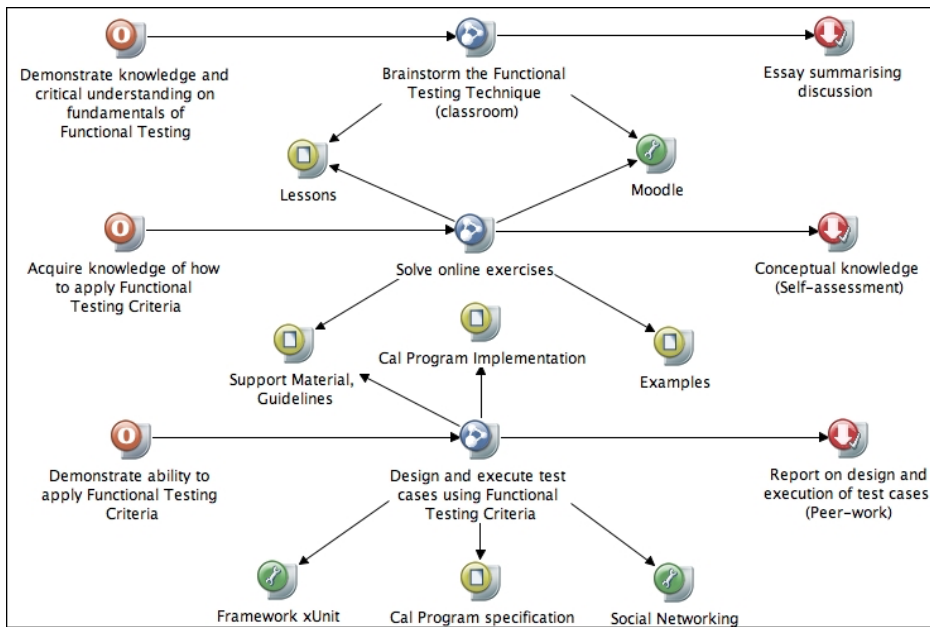


Fig. 5. OER on software testing: learning structure overview.

During the course development, the learning structure was refined to visualise the flow of course activities and to determine the tasks that students need to perform to achieve the intended learning outcomes. A sketch of this refinement is represented by the Learning Tasks View (Fig. 6). A simple learning activity (*design and execute*

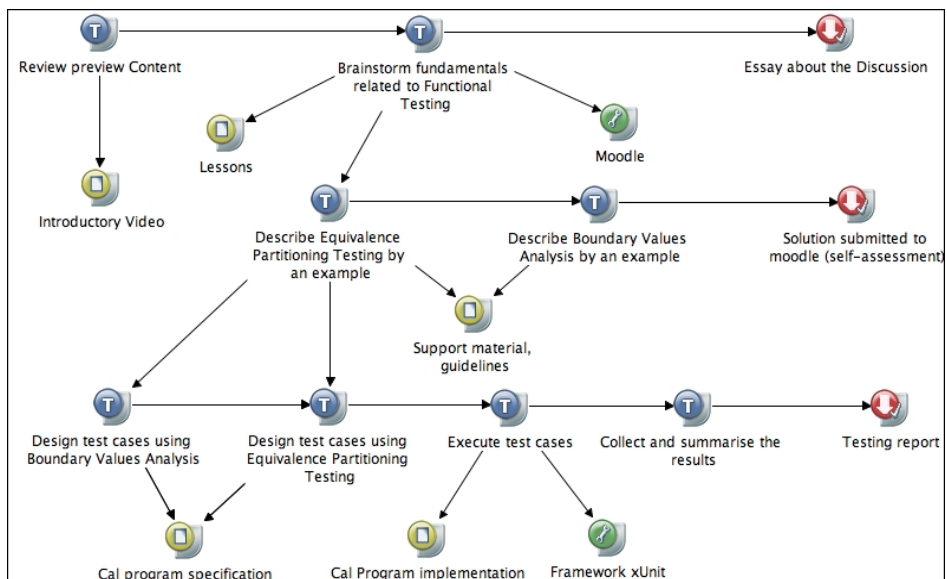


Fig. 6. OER on software testing: learning structure refinement.

test cases using functional testing criteria) from the Learning Structure View was broken down into simple and atomic tasks (such as *design test cases using boundary values analysis* and *execute test cases*) together with related contents, tools and assessments.

In comparison to the AD-HOC method, the definition and mapping of the learning structure and activities of the course through the Learning Structure View and Learning Tasks View represent activities specifically related to the AM-OER method.

The contents needed for the course were created, reviewed and integrated with other components of the course. Contents include html pages/text documents, lessons/lecture notes, images and video. In Fig. 7 we illustrate one of the learning assessment activities proposed for students, consisting of a practical activity involving the design and implementation of partitioning functional criteria.

At the end of the project, all components of the course were evaluated by the educator and learners within the group. The evaluation activities were performed according to the criteria/aspects considered in the AM-OER method. Feedback obtained was gathered and used to improve the whole course.

The OER on software testing course developed are available under the Creative Commons License – Attribution Non-Commercial Share Alike 4.0 International (Creative Commons, 2013). The course was provided to graduate students in Computer Science at the University of São Paulo (ICMC/USP).

The data collected and feedback obtained from participants were also used to refine and improve the latest version of AM-OER.

The screenshot displays a web-based assessment activity interface. At the top, there is a navigation bar with a 'Menu' button on the left and '« Previous' and 'Next »' buttons on the right. Below the navigation bar, the word 'Test' is prominently displayed in green. On the left side, there is a sidebar menu titled 'SOFTWARE TESTING' with several sub-items: 'Software Testing Techniques', 'Functional Testing Criteria', 'Partitioning Testing', 'Boundary Values Analysis', 'Test' (highlighted in red), and 'Summary'. The main content area is titled 'Partitioning Testing' and contains a 'String program specification' section. This section describes a program that takes a positive integer (1 to 20) and a string as input, returning the position of the first occurrence of the character or a message if not found. Below the specification, there are instructions for the user to identify equivalence classes and define test cases, with a deadline of 24 hours. At the bottom of the main content area, there is a section titled 'Source code for String program' which includes a link to 'string.c (New Window)'.

Fig. 7. OER on software testing: assessment activity.

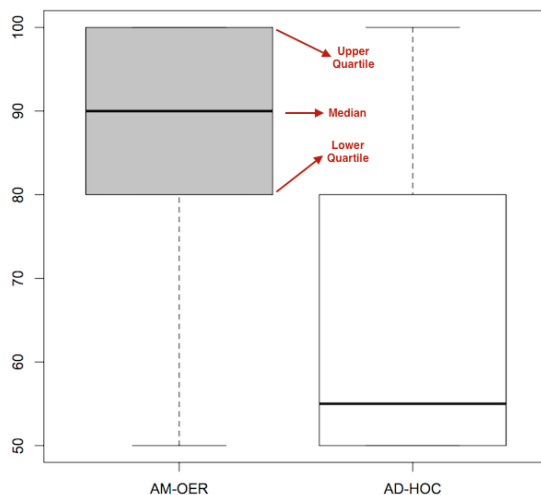


Fig. 8. Result obtained by each method.

5. Analysis and Results

In Fig. 8 we show the results obtained by each group using their respective method, displayed by box-plots representing the sample data in three quartiles:

- (1) Lower Quartile (Q1): the value representing a quarter of the sample data.
- (2) Median Quartile (Q2): the value representing the median of the sample data.
- (3) Upper Quartile (Q3): the value representing three quarters of the sample data.

The box-plots also show the minimum and maximum values of the sample, ranging from 50 to 100%. In the sample, AM-OER achieved a better result between 80 and 100%.

5.1. RQ1. How Effective is AM-OER in the Development of OERs Compared to the AD-HOC Method?

RQ1 aimed at assessing the effectiveness of AM-OER compared to the AD-HOC method. It is related to the OER module “planned” to be developed and to the OER module actually “developed” and “delivered” at the end of the experiment.

Effectiveness is given by $\sum (x_i/y_i) * 100$, $i = 1..n$, where x_i represents the average percentage of requirements fulfilled by each method, whilst y represents the requirement planned to be fulfilled.

The results show that AM-OER obtained 86.2% against 65% of the AD-HOC method. We intended to test the hypotheses for effectiveness. As we are working with proportional sampling (both sample are presented as percentage), test for proportion (πp) is indicated. To test both samples for equality proportions we used the R software (R Project, 1993).

By admitting a low value or level of significance as $\alpha = 0.5$ at a 95% percentage confidence interval, the result was $p\text{-value} = 0.000882$. There is a statistical significance when the $p\text{-value}$ is lower than the level of significance (α) established for the study.

From the results, we rejected the null hypothesis $\rightarrow H_0: \pi\text{AM-OER}_{\text{Effe}} = \pi\text{AD-HOC}_{\text{Effe}}$ and accepted the alternative hypothesis $\rightarrow H_1: \pi\text{AM-OER}_{\text{Effe}} > \pi\text{AD-HOC}_{\text{Effe}}$. Then, we can statistically infer that the effectiveness of AM-OER is greater than the effectiveness of the AD-HOC method.

5.2. RQ2. How Efficient is AM-OER in the Development of OERs Compared to the AD-HOC Method?

RQ2 aimed at assessing the efficiency of AM-OER compared to the AD-HOC method. It is related to the OER module “developed” and the “time” spent to do the work. Actually, the results of RQ1 and RQ2 are dependent on each other. To obtain RQ2 we used the results from RQ1 plus the time.

Efficiency is given by $\sum (xi/yi)$, $i = 1...n$, where xi represents the average percentage of requirements fulfilled by groups on each method, whilst y represents the time (in hours) spent by them.

The results for efficiency show that AM-OER obtained 0.86 against 0.73 of AD-HOC. The higher value obtained, the greater the efficiency. We intended to test the hypotheses established for efficiency. With a level of significance of $\alpha = 0.5$ at a 95% percentage confidence interval, the obtained result was $p\text{-value} = 0.03556$.

From the results, we rejected the null hypothesis $\rightarrow H_0: \pi\text{AM-OER}_{\text{Effi}} = \pi\text{AD-HOC}_{\text{Effi}}$ and accepted the alternative hypothesis $\rightarrow H_1: \pi\text{AM-OER}_{\text{Effi}} > \pi\text{AD-HOC}_{\text{Effi}}$. Then, we can statistically infer that the efficiency of AM-OER is greater than the efficiency of the AD-HOC method.

5.3. RQ3. How Much Better are the Results Obtained by AM-OER Compared to the AD-HOC Method?

RQ3 aimed at assessing the quality of the results obtained by the groups on each method. This is carried out by a specialist from the University of São Paulo (USP), taking into account the percentage of compliance to quality attributes of OERs. They are derived from the main characteristics of an OER such as:

- (1) Contents fitness for purpose.
- (2) Contents accurate and readable.
- (3) Adherence to open technical standards.
- (4) Metadata description.
- (5) Accessible and modifiable formats.
- (6) Facility for repurpose and reuse.
- (7) Well-defined open licensing policies, among others.

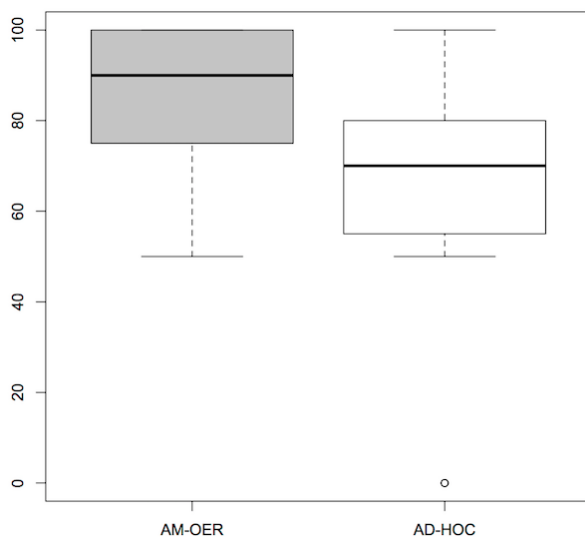


Fig. 9. Quality of the results of each method.

In Fig. 9 we show the percentage obtained by each method through box-plots. The box-plot of AM-OER shows the results are closer to 80% and 100%, ranging from 50% (minimum value) to 100%. On the other hand, the box-plot of AD-HOC shows a higher variation, ranging from 0% to 100%.

We intended to test the hypotheses for quality results. With a level of significance of $\alpha = 0.5$ at a 95% percentage confidence interval, the result was $p\text{-value} = 0.0006052$.

From the results, we rejected the null hypothesis $H_0: \pi_{AM-OER_{Qual}} = \pi_{AD-HOC_{Qual}}$ and accepted the alternative hypothesis $H_1: \pi_{AM-OER_{Qual}} > \pi_{AD-HOC_{Qual}}$. Then, we can statistically infer that the quality of the results obtained by AM-OER is greater than the quality of the results obtained by the AD-HOC method.

5.4. Applicability of AM-OER

To explore the applicability of AM-OER in the development of OERs we investigated a set of research questions covering three perspectives: (1) appropriateness/usefulness: the level in which the method offers adequate support and guide the development of OERs; (2) ease of use: the level in which the method is easy to be used, especially by non-specialist in the development of OERs; and (3) satisfaction: the level in which the participants wish to use the method in future projects and would recommend it.

According to the results, all answers for the research questions covering the perspectives aforementioned were between “4 – Partially agree” and “5 – Strongly agree” (in a scale from “1 – Strongly disagree” to “5 – Strongly agree”). The results show a tendency of the acceptance of AM-OER in the development of OERs. However, other assessments must be conducted in order to provide more consistent results.

The participants provided suggestions for changes and enhancements to the AM-OER method, particularly regarding the identification of the learning needs for the course. They argued that this step was not enough clear, requiring a more objective and detailed explanation on how to perform it. The participants also reported the difficulty and effort needed to find related contents with open and/or compatible licenses, which may hinder the reuse third-part materials during the development process.

The data collected and feedback obtained from participants were used to refine and improve the latest version of AM-OER.

5.5. Threats to Validity

Some issues that may affect and impact the analysis and interpretation of the results were:

- **Learning:** the participants may have difficulty in using AM-OER method. We have tried to minimize this effect by conducting an appropriate training covering the main characteristics, practices and activities of AM-OER method. All training materials were shared with the participants. Besides that, they received all instructions needed to develop their tasks.
- **Conformance to the original study:** there may be a discrepancy between the training step and the original study. We have carefully planned the time of the training development tasks in accordance with the time needed for the development tasks of the experiment.
- **Environment of the experiment:** the environment must be suitable for the experiment. We conducted the experiment in laboratory with no external interference and adequately furnished with computer and Internet access.
- **Experience of participants:** the level of experience of participants could influence in the validation. Before the experiment, participants were asked to answer a questionnaire about their level of experience, covering the subjects involved in the experiment. From this, we have tried to create homogeneous groups with similar levels of experience on the subjects.
- **Time for the experiment:** the time allocated to the experiment was short due to the unavailability of participants, usually involved with other activities. In this case, we established a simple and short module suitable to be developed in the experiment. Only one five hours sprint could seem as the application of waterfall (Royce, 1970) instead of an agile method. However, in the sprint of our experiment, all phases were performed together, with the active participation and collaboration between the participants. The module was defined, designed, created and evaluated during the sprint and, then, delivered to a target audience.
- **Number of participants:** the number of participants of the experiment is relatively small and may not adequately reveal the applicability and effectiveness of the AM-OER to support the development of OERs. Currently, we are working to replicate and plan new experiments with a large number of participants.

6. Conclusions and Future Work

In this paper we established and applied an agile method for the design and creation of OERs (AM-OER). The AM-OER method differs from other general approaches for the development of learning materials, by proposing and adapting agile practices from Software Engineering in combination with Learning Design practices from OULDI to the context of OERs development. Pedagogical design practices are embedded in the OERs development process, improving quality and facilitating reuse and adaptation.

AM-OER aims to fill the gap regarding the lack of systematic and flexible approaches focused on the development of OERs (Petrides *et al.*, 2008; Sclater, 2009; OPAL, 2011; Arimoto and Barbosa, 2012). AM-OER promotes flexibility for accommodating changes and improvements throughout the development. An OER can be modified, repurposed and evolved according to the needs that emerge during the development, minimizing its cost and impact. AM-OER prioritizes the cooperation and collaboration of those involved in the process. The active participation of educators and learners is encouraged throughout the process. Such characteristics help to reduce the development time and effort, promoting an effective production process for a sustainable supply of OERs.

The empirical assessment conducted to validate AM-OER has shown the effectiveness and efficiency of the method in the design and creation of OERs. The obtained results also have shown the method is useful and easy to use. Despite the lack of experience of some participants, the results indicated the AM-OER can be used by non-specialists in the development of OERs.

In the experiment we have only compared AM-OER with an AD-HOC method because our first intention was to assess the feasibility and applicability of the proposition of an agile method for the development and delivery of OERs. As future work, we are planning to compare AM-OER with more systematic methods, including plan-driven methods such as waterfall (Royce, 1970).

The experiment considered a small number of participants, which may affect the representativeness of the sample data. New empirical assessment studies with a large number of participants should be conducted to validate AM-OER, including the development of OERs applied to other learning contexts. The idea is to develop and provide OERs for teaching and training in a variety of areas and knowledge domains. A theoretical framework to support the assessment of OERs should be defined as well.

Acknowledgment

We are grateful to the Brazilian funding agencies FAPESP (grants 2011/06204-1 and 2013/16759-6), CNPq and CAPES for their financial support.

References

- Ambler, S.W. (2001). Agile design practices. agilemodeling.com/essays/agileDesign.htm
- Arimoto, M.M., Barbosa, E.F. (2012). A systematic review of methods for developing open educational resources. In: *20th International Conference on Computers in Education (ICCE' 12)*. Singapore, 262–269.
- Arimoto, M.M., Barroca, L., Barbosa, E.F. (2014). Recursos educacionais abertos: Aspectos de desenvolvimento no cenário brasileiro. *Revista Novas Tecnologias na Educação (RENOTE)*, 12(2), 1–10.
- Avraamidou, A., Economou, A. (2012). Visualized learning design: The challenges of transferring an innovation in the Cyprus educational system. *Teaching English with Technology, Special Edition on LAMS and Learning Design*, 12(3), 3–17.
- Basili, V.R., Forest, S.F.L. (1999). Building knowledge through families of experiments. *IEEE Transactions on Software Engineering*, 25(4), 456–473.
- Beck, K., Andres, M. (2004). *Extreme Programming Explained: Embrace Change*. Addison-Wesley.
- Bloom, B., Englehart, M., Furst, E., Hill, W., Krathwohl, D. (1956). *Taxonomy of Educational Objectives: The Classification of Educational Goals*. Longmans, Green, Toronto. Handbook I: Cognitive domain.
- Boyle, T., Cook, J., Windle, R., Wharrad, H., Jeeder, D., Alton, R. (2006). An agile method for developing learning objects. In: *23rd Annual Ascilite Conference: Who's Learning? Whose Technology?*, Sydney, Australian, 189–196.
- Braga, J.C., Pimentel, E., Dotta, S. (2013). Metodologia INTERA para o desenvolvimento de objetos de aprendizagem. In: *XXIII Simpósio Brasileiro De Informática na Educação*, 306–315.
- Brasher, R., Cross, A., Galley, S., Weller, M. (2012). Final project report of the OULDI-JISC Project: Challenge and Change in Curriculum Design Process.
- Conole, G. (2013). Designing for learning in an open world (*Explanation in the Learning Sciences, Instructional Systems and Performance Technologies Series*). Springer.
- Conole, G., Wilson, T., McAndrew, P., Culver, J., Brasher, A., Cross, S. (2009). New methods and approaches to the design and evaluation of OERs. In: *EDEN Annual Conference 2009*. Gdansk, Poland.
- Creative Commons (2013). About the Creative Commons Licenses. <http://creativecommons.org/licenses/about>
- DeVries, I. (2013). Evaluating open educational resources: lessons learned. *Procedia – Social and Behavioral Sciences*, 83, 6–60.
- Dimitriadis, Y., McAndrew, P., Conole, G., Makriyannis, E. (2009). New design approaches to repurposing OERs for collaborative learning using mediating artefacts. In: *Auckland: ASCILITE*.
- International Organization for Standardization (2001). Software Engineering – Product quality – Part 1: Quality model. *Standard ISO/IEC 9126-1:2001*, ISO/IEC.
- JISC. (2010). Open Educational Resources (OERs). https://jisc.ac.uk/guides/open_educational-resources.
- Kemczinski, A., Costa, I.A., Wehrmeister, M.A., Hounsell, M.S., Vahldick, A. (2012). Metodologia para construção de objetos de aprendizagem interativos. In: *XXIII Simpósio Brasileiro de Informática na Educação*, 1–10.
- Lapolli, F., Motta, C.L.R., Cruz, C.M., Oliveira, C.E.T. (2010). Modelo de desenvolvimento de objetos de aprendizagem baseado em metodologias ágeis e scaffoldings. *Revista Brasileira de Informática na Educação*, 18(2), 41–53.
- Leacock, T.L., Nesbit, J.C. (2007). A Framework for evaluating the quality of multimedia learning resources. 10(2), 44–59.
- Levery, L. (2012). Finding relevant OER in higher education: a personal account. In: Glennie, J., Harley, K., Butcher, N. and Wyk, T. van (Eds), *Open Educational Resources and Change in Higher Education: Reflections from Practice*.
- Madden, T. (2010). Turning a resource into an open educational resource (OER). *UK Physical Sciences Centre – Briefing Paper*.
- Masson, P., Udas, K. (2009). An agile approach to managing open educational resources. *On the horizon*, 17(3), 256–266.
- Molenda, M. (2003). In search of the elusive ADDIE Model. *Performance Improvement*, 45(5), 34–36.
- Myers, G.J. (2004). *The Art of Software Testing*. John Wiley & Sons, New York, 2 edition.
- Open Educational Quality Initiative (2011). Beyond OER: Shifting focus to open educational practices. *OPAL Report 2011*. Essen, Germany: Open Education Quality Initiative.
- Organisation for Economic Co-operation and Development (2007). *Giving Knowledge for Free. The Emergence of Open Educational Resources*. Centre for Educational Research and Innovation (CERI).

- Patricia, C.C.S., Rocio, R.M.G. d., Elizabeth, R.P.A. (2010). OER's production cycle with social authorship and Semantic Tools. In: *IEEE Education Engineering Conference, EDUCON 2010*, 121–128.
- Petrides, L., Nguyen, L., Kargliani, A., Jimes, C. (2008). Open educational resources: inquiring into author reuse behaviours. In: P. Dillenbourg & M. Specht (Eds.), *Times of Convergence. Technologies Across Learning Contexts*. Springer, Berlin, 44–353.
- Plackett, R.L. (1983). Karl pearson and the chi-squared test. *International Statistical Review*, 51(1), 59–72.
- R Project (1993). The R Project for Statistical Computing. <https://www.r-project.org/>
- Schuwert, R., Wilson, T., Valkenberg, W.V., Lane, A. (2010). Production of OER: a quest for efficiency. In: *7th Annual Open Education Conference*. Barcelona, Spain, 2–4.
- Schwaber, K. (1995). The scrum development process. In: *Proceedings of the 10th Annual ACM Conference on Object Oriented Programming Systems, Languages, and Applications (OOPSLA'95)*. Austin, Texas, USA, 117–134.
- Sclater, N. (2009). The organizational impact of open educational Resources. In: U.-D. Ehlers & D. Schneckenberg (Eds.), *Changing Cultures in Higher Education: Moving Ahead to Future Learning*. 1–14.
- Shapiro, S.S., Wilk, M.B. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4), 591–611.
- Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslen, A. (2000). *Experimentation in Software Engineering: An Introduction*. Kluwer Academic Publishers, Massachusetts, USA.
- Yuan, L., Mac Neill, S., Kraan, W. (2008). Open educational resources – opportunities and challenges for higher education. *JISC CETIS*.

M.M. Arimoto is a professor at the Centre of Technological Sciences – State University of Northern Paraná (UENP) (Brazil), with a PhD in Computer Science and Computational Mathematics from the Institute of Mathematics and Computer Sciences – University of São Paulo (ICMC-USP) (Brazil). During his PhD, he was an academic visitor at the Faculty of Mathematics, Computing and Technology – The Open University (United Kingdom). His research interests are focused on CS and SE education and training, open educational resources, learning design, agile methods, software quality, and open source software.

L. Barroca is a senior lecture in Computing at the Faculty of Mathematics, Computing and Technology – The Open University (United Kingdom), with a PhD in Computer Science from Southampton University, and an Msc in Computing from University of Oxford. She has been involved with teaching and research in software engineering and with the design of, and use of technology for, distance education in software engineering.

E.F. Barbosa is an Associate Professor at the Institute of Mathematics and Computer Sciences – University of São Paulo (ICMC-USP), Brazil. She received her BS in Computer Science in 1995 from the State University of Londrina (UEL), Brazil, her MS in Software Engineering in 1998 from ICMC-USP, and her DS in Software Engineering in 2004, also from ICMC-USP. During her DS, she was a visitor scholar at Georgia Institute of Technology and at University of Florida, United States. Her main research interests are related to CS and SE education and training, open education resources, mobile learning, educational data mining, software testing, empirical software engineering, and software product lines. A list of her main publications can be found on Lattes Platform: <http://lattes.cnpq.br/7913302545613108>

